A Comparison of Two Path Planners for Planetary Rovers

M. Tarokh[1], Z. Shiller[2] and S. Hayati[3]
1 Robotics and Intelligent Systems Laboratory
San Diego State University, San Diego, CA 92182-7720
2 Dept of Mechanical and Aerospace Engineering
University of California, Los Angeles, CA 90095
3 Jet Propulsion Laboratory
California Institute of Technology, Pasadena, CA 91109

Abstract

The paper presents two path planners suitable for planetary rovers. The first is based on fuzzy description of the terrain, and genetic algorithm to find a traversable path in a rugged terrain. The second planner uses a global optimization method with a cost function that is the path distance divided by the velocity limit obtained from the consideration of the rover static and dynamic stability. A description of both methods is provided, and the results of paths produced are given which show the effectiveness of the path planners in finding near optimal paths. The features of the methods and their suitability and application for rover path planning are compared.

1 Introduction

Following the successful launch and deployment of Mars Sojourner rover, NASA has planned further rover missions to Mars starting in 2001 with Marie Curie, a rover similar to the Sojourner. Two additional rover missions in 2003 and 2005 have been planned for in-situ experiments, and another in 2007 for sample return to Earth. An important element for the success of these missions is incorporating a reasonably high level of autonomy in the rover so that it can traverse distances of 100 meters or more per communication cycle. In order to traverse these distances, it is necessary to delegate the motion planning task to the rover using the image obtained from mast mounted cameras. The challenge is then to use these images to perform on-board path planning.

The existing path planners focus almost exclusively on obstacle avoidance, treating obstacles as forbidden regions and the rest of the terrain as free spaces [1]. This binary environment is not appropriate for the Martian terrain and a rover that can climb over some rocks [2] if such traversals result in more optimal routes. In fact NASA's experience with Sojourner has revealed many cases where a binary obstacle model has resulted in halted motions, often leaving the rover in an undesirable situation [3]. Recently several path planners have been developed that consider the traversability of the terrain [4]-[7]. Terrain topology and simple vehicle dynamics are considered in [4] to generate global optimal paths on general terrain. In [5] the shortest feasible path for off-road vehicles is computed. A genetic algorithm is used in [6] to synthesize path from segments, each evaluated for its static stability and for satisfying certain mission tasks. A recently developed planner [7] uses fuzzy logic to characterize the terrain traversability, and then finds traversable paths in a rocky terrain.

The purpose of this paper is to discuss two path planners for possible Mars rover applications. The first algorithm is based on fuzzy characterization of the terrain roughness, and the use of a genetic planner to optimize a fitness function. The second algorithm considers constraints imposed by certain vehicle dynamics and terrain topology to come up with an optimal path. The common feature of both planners is finding paths that are optimal in the sense of both distances and traversability, where the latter quantifies the ease of traversal of the terrain. These two algorithms find paths that result in reduced rover energy consumption and enable exploring larger regions of the Martian terrain.

2 Genetic Path Planner

The path planner starts by creating several random paths between start and goal points on the terrain. These initial paths in general go though rough or impassable regions on the terrain, and must be improved. This improvement is achieved by applying certain genetic operators to a randomly selected path from the population. Each genetic operator has a particular role in bringing about a change in the path. For example, *replace* operator replaces an undesirable way-point (a way-point on a rough region), with a random and potentially better way-point. The selection of particular operator is based on the probability assigned to it. After a genetic operation is performed, the quality of all paths are compared, and the worst path is eliminated from the population. The process of applying a genetic operator to create a new path, and eliminating the worst path, is referred to as a generation. The population goes through generations and is thus evolved. After each generation, the quality of the paths is either improved or in the worst case remain unchanged. The evolution is continued until an acceptable path is found, or until a preset number of generations are performed.

2.1 Terrain Roughness

Consider a terrain divided into a grid of regular square cells whose size depends on the dimension of the rover, and the desired resolution of surface description. The roughness of a flat obstacle free cell is assigned a value of 0, and that of a rugged cell with large obstacles is assigned a value of 1. The measure of roughness depends on a number of parameters as follows:

- Height of the tallest obstacle in the cell - The roughness becomes smaller with a decrease in the rock height.
- Size or surface area of the cell occupied by obstacles or rocks - If two cells have rocks of the same height, the region with less rock occupied area is smoother and thus has a lower roughness value.

In addition to roughness, two path dependent quantities, namely path slope and curvature, affect the difficulty of the traversal by a rover. These will be considered in Section 2.2.

The most commonly used sensors for mobile robots are cameras and their associated image processing hardware and software. Despite the availability of vision processing software, exact determination of the heights and sizes of rocks affecting roughness is not possible. These parameters can be found, at best, approximately due to errors, misinterpretations and ambiguity involved in extracting information from images. It is therefore essential to set the problem in a fuzzy and approximate reasoning framework.

The height of the tallest rock in the cell under consideration, h, and the size or surface area occupied by rocks in this cell, s, are used to find the cell roughness ρ. The crisp values of h, s and ρ are fuzzified to obtain the linguistic variables $\tilde{h}, \tilde{s}$ and $\tilde{\rho}$, respectively. The "if-then rule" of the following form is employed to obtain the fuzzy roughness,

$$if\ \tilde{h}\ is\ \tilde{H}_k\ and\ \tilde{s}\ is\ \tilde{S}_k\ \ then\ \tilde{\rho}\ is\ \tilde{\rho}_k \qquad (1)$$

where $\tilde{H}_k$, $\tilde{S}_k$ and $\tilde{\rho}_k$, $k = 1, 2, \cdots, \nu$ are the linguistic values associated with $\tilde{h}, \tilde{s}$ and $\tilde{\rho}$, respectively, and ν is the number of linguistic values. The fuzzy sets H_k, S_k and ρ_k are used to quantify the linguistic statements "$\tilde{h}$ is $\tilde{H}_k$", "$\tilde{s}$ is $\tilde{S}_k$" and "$\tilde{\rho}$ is $\tilde{\rho}_k$", respectively. The fuzzy sets H_k for the hight are chosen as very low ($H_1 \equiv VL$), low ($H_2 \equiv LO$), medium ($H_3 \equiv ME$), high ($H_4 \equiv HI$) and very high ($H_5 \equiv VH$). The membership functions $\mu_{H\,k}$ for these fuzzy sets are standard triangular and have equal base width with a 25% overlap. The fuzzy sets associated with the rock size are tiny ($S_1 \equiv TI$), small ($S_2 \equiv SM$), medium ($S_3 \equiv ME$), large ($S_4 \equiv LG$) and extra large ($S_5 \equiv XL$), and are also triangular with 25% overlap. The fuzzy sets for roughness are very low ($\rho_1 \equiv VL$), low ($\rho_2 \equiv LO$), medium ($\rho_3 \equiv ME$), high ($\rho_4 \equiv HI$) and very high ($\rho_5 = VH$). The membership functions $\mu\rho_k$ for the roughness are designed to be triangular with different base widths to give more weighting to rougher terrains.

The rule matrix implementing (1) is given in Figure 1, and consists of 25 rules which are self-explanatory. Zadeh's compositional rule of inference, and center of height defuzzification method is used to obtain the crisp value of the cell roughness ρ.

2.2 Path Representation

A path is represented by a sequence of way-points connecting the start to the goal. The way-points W_k, $k = 1, 2, \cdots, m$ are specified by their (x_k, y_k) coordinates on the terrain. The generation and evolution of a path refers to the creation and modification of the way-points. These way-points in turn specify the terrain cells that the path traverses over. A cell that is located on a path, will be referred to as a *path cell*, and has two main attributes as follows:

- The roughness ρ_i of the cell, which provides information on the heights, sizes and concentration of rocks on a cell, as described in Section 2.1.
- The curvature or jaggedness of a path cell is obtained using the information about the way-points. Specifically, the curvature $\hat{\zeta}_k$ of the way-point W_k is defined as

$$\hat{\zeta}_k = \frac{d_k}{D_k} \qquad k = 1, 2, 3 \cdots, m \qquad (2)$$

where d_k is the perpendicular distance of W_k to the line segment joining the previous way-point W_{k-1} to the next way-point W_{k+1}, and D_k is the distance between W_{k-1} and W_{k+1}. Note that $\hat{\zeta}_k$ is a dimensionless quantity, and that $0 \leq \hat{\zeta}_k < \infty$. Furthermore, (2) also gives the curvature of the path cell that contains a way-point.

It is noted from Section 2.1 that roughness is normalized and varies between 0 and 1. However, curvature can have large values. In order to enable easy comparison between the two cell attributes, we normalize curvature as follows:

$$\zeta_i = 1 - e^{-a\hat{\zeta}_i} \tag{3}$$

where a is a constant whose role will be explained shortly. Note that $0 \leq \zeta_i \leq 1$ for all values of a.

The above two quantities, namely roughness and curvature, which are attributes of path cells, are combined to define a *cell impedance* η_i as follows

$$\eta_i = \frac{1}{2}(\rho_i + \zeta_i) \tag{4}$$

The cell impedance varies between 0 and 1 and quantifies the difficulty of the path cell traversal by a rover. Consequently, a path cell containing no rocks that is located on a straight path segment will have a minimum impedance of 0. On the other hand a very rough cell on a jagged path segment will have a maximum impedance of 1. The constant a in (3) determines the weight given to curvature relative to the roughness. Lower values of a reduce the contribution of curvature to the overall cell impedance. It is noted that other path attributes such as slope can easily be included in the above formulation of the path impedance.

A cell with an impedance of more than a threshold becomes *intraversable.* The value of the threshold is chosen based on the mobility characteristics of the particular rover being used. We identify a path as being traversable if every cells on the path is traversable, otherwise the whole path becomes intraversable. In the genetic evolutionary process, these two type of paths are treated separately. Although, traversable paths have priority over intraversable paths, the latter are not automatically discarded since they may prove to produce good offsprings later on during the evolutionary process. The *path impedance* is defined as the sum of impedances of all cells on the path, that is

$$\eta = \sum_{k=i}^{n} \eta_i \tag{5}$$

When a population of paths consisting of both traversable and intraversable paths are compared for selection, any traversable path is given preference over best (lowest η) intraversable path. However, when the population consists of only traversable paths or only intraversable paths, then the selection is based on lower values of η.

2.3 Genetic Operators

In order to evolve paths from one generation to the next, several operators have been devised. Two of these operators, namely cross over and mutation, are commonly used in genetic algorithms. Others are specifically designed for the path planner. Operators are applied to way-points, and as a results of changes in way-points, the path cells are also changed. Note that each time an operator is applied, a new path is generated. If this new path produces a path impedance that is lower than the impedance of any path in the population, it is accepted as a new member of population, and the path with highest impedance is discarded.

Cross-Over

This operator randomly selects two paths from the population, say P_1 and P_2, and divides each path into two path segments about a randomly elected way-point. Denoting these paths by $P_1 = (P_{11}, P_{12})$ and $P_2 = (P_{21}, P_{22})$, where P_{ij} is the j-th segment of path i, then two new paths are formed as $\hat{P}_1 = (P_{11}, P_{22})$ and $\hat{P}_2 = (P_{21}, P_{12})$.

Mutate

This operator randomly selects a path and a way-point in this path. It then changes the x, y coordinates of the selected way-point with random values. Mutate operator can produce a significant change in the path.

Replace

This operator is applied to an intraversable path. It replaces an intraversable way-point with one or more way-points whose location and number are random. If there are more than one intraversable way-points, one of them is selected randomly for replacement.

Swap

The operator interchanges the locations of two randomly selected way-points on a randomly selected path. The swap operator can be applied to both traversable and intraversable paths. It has the possibility of removing or introducing a "zig-zag".

Smooth

The role of this operator is to reduce sharp turns. The way-point with the highest curvature, say W_k, is selected and two new way-points are inserted, one on a randomly selected cell between the way-points W_{k-1} and W_k and the other on a cell between W_k and W_{k+1}. After this insertion, the way-point W_k is removed. The effect of this operation is the smoothing of a sharp turn. This operator is only applied to traversable paths.

Pull-out

This operator is intended to pull out a path segment from inside an intraversable region to its surrounding traversable region. Pull-out is more elaborate than the other operators, and details of its implementation is omitted here for the sake of brevity.

The probability of occurrence of an operator depends on the role played by it in the evolution of paths. An adaptation scheme is devised to modify the probabilities based on the population diversity, and traversability. For example, if most paths in the population are similar and have high impedances, mutation is given higher probability and cross over is assigned a smaller probability. This is due to the fact that in this situation cross over of intraversable paths also produce other intraversable paths and a substantial change is needed which is achieved by mutation.

3 The Global Optimization Planner

This planner formulates the motion planning problem as a three stage optimization. At the lowest level, a given path is evaluated for its traversability by computing the maximum speeds along the path at which the vehicle is dynamically stable. The second level consists of a parameter optimization that selects a locally optimal path in the neighborhood of an initial guess. The third and highest level of the optimization selects the initial guesses for the local optimization. The global optimization is based on a branch and bound search that prunes the initial set of all paths between the end points to a small number of candidates for the local optimization [4]. These candidates represent the most promising regions, one of which contains the global optimal path. Optimizing these paths with the local optimization yields the best path, in addition to a number of good alternatives. These paths are not necessarily the shortest, but they are traversable at the widest speed range of all paths with similar or shorter lengths, as is demonstrated in several examples in this paper.

3.1 Terrain and Path Representation

The terrain is represented by a cubic B patch, which is a parametric surface made of a mesh of cubic splines. A typical point p on a single patch in three dimensional space is a function of two parameters, v and w, :

$$p = VMRM^TW^T \qquad (6)$$

where $V = [v^3, v^2, v, 1]$, $v = [0,1]$, $W = [w^3, w^2, w, 1]$, $w = [0,1]$ M is the 4×4 matrix specifying the type of spline used to construct the patch, and R is a 4×4 matrix of 16 control points.

The control points of the patch are generated by placing a uniform grid on the map-range data generated from stereo images taken by the on-board mast camera. The resolution of this grid is chosen economically at about half the rover size: roughly $20cm$ between neighboring points. This ensures that obstacles the size of the rover and larger are depicted by the B-patch. Smaller obstacles may be filtered out.

The path is represented by a smooth curve on the surface, obtained by parameterizing v and w by a single parameter u:

$$c(u) = p(v(u), w(u)) = V(u)MRM^TW^T(u) \qquad (7)$$

Reducing the $v - w$ space to a line reduces the B patch to a continuous curve that is guaranteed to stay on the surface.

3.2 Vehicle Model

At top speeds of 10 - 20 cm/s, the motion planning problem for Mars Rover can be considered a kinematic problem. However, we do account for certain rover dynamics for the purpose of quantifying traversability and dynamic stability, with the premise that paths that are traversable at a wide speed range are safer than those that are not.

The vehicle is modeled as a point mass, suspended above ground at the location of the vehicle's center of mass. The height of the center of mass above ground and the width between the wheels are used to evaluate stability with respect to lateral tip over.

The external forces acting on the vehicle consist of the friction force F (the sum of all the horizontal tire forces), the normal force R (the sum of all normal tire forces) applied by ground on the vehicle in the r direction, and the gravity force.

The equation of motion of the vehicle are written in the vehicle fixed frame in terms of the tangential speed $\dot{s}$ and the tangential acceleration $\ddot{s}$ [4]

$$\begin{aligned} f_t &= mgk_t + m\ddot{s} & (8)\\ f_q &= mgk_q + m\kappa n_q \dot{s}^2 & (9)\\ R &= mgk_r + m\kappa n_r \dot{s}^2 & (10) \end{aligned}$$

where f_t and f_q are the components of the friction force tangent and normal to the path, k_t, k_q and k_r are the projection of the vertical unit vector, k, on the respective axis of the vehicle fixed coordinate frame, and $1/\kappa$ is the path curvature, . The moment of the friction force around the center of mass is considered later when we account for the tip over constraint.

Equations (8) to (10) are used to determine the feasible speed and acceleration for given limits on the friction and normal forces.

3.3 Dynamic Constraints

Constraints between the vehicle and ground are considered to ensure vehicle dynamic stability along the path.

Sliding Constraint

The maximum friction force is a function of the normal force and the coefficient of friction between the wheels and ground:

$$|F| \leq \mu R \qquad (11)$$

Substituting (8)- (10) in (11), then solving for $\ddot{s}$ yields constraints of the form [4]

$$-gk_t + \sqrt{\Delta} \leq \ddot{s} \leq -gk_t - \sqrt{\Delta} \qquad (12)$$

where

$$\Delta = a\dot{s}^4 + 2b\dot{s}^2 + c \geq 0 \qquad (13)$$

yields constraints on the feasible vehicle speed along the path. The feasible speed range is determined by the roots of (13). Only the positive roots are of interest.

Contact Constraint

To ensure that the vehicle does not loose contact with ground on rough terrain, the normal force R applied on the vehicle should be positive. 'Setting $R = 0$ in (10), we obtain the maximum speed allowed by the contact constraint:

$$\dot{s} \leq \sqrt{-\frac{gk_r}{\kappa n_r}}. \qquad (14)$$

where n_r is the projection of the path normal, n, on the surface normal, r. Equation (14) applies only for the cases where path curvature points opposite to the direction of the surface normal. Note that the velocity limit is infinite for a flat terrain ($n_r = 0$), and zero for a sharp vertical bump ($\kappa n_r = \infty$), as expected.

Tip-Over Constraint

The tip-over constraint is obtained by expressing the limiting condition before the vehicle is about to tip-over in terms of $\dot{s}$, $\ddot{s}$. The vehicle will not tip-over if the reaction force and the lateral friction force satisfy [6]

$$f_q^2 \leq (R\frac{b}{h})^2 \qquad (15)$$

Substituting (8) and (9) into (15) yields a constraint on $\dot{s}$ similar to (13).

Velocity Limit Curve

Plotting the velocity limits due to the dynamic constraints along the path forms the *velocity limit curve* in the phase plane $s - \dot{s}$. It represents the upper bound for vehicle speeds for which the dynamic constraints discussed earlier are satisfied. The height of the velocity limit represents a measure of safety and traversability: a zero velocity limit implies static instability, whereas a nonzero but low velocity limit implies a stable but dangerous position along the path. Obviously, the higher the velocity limit, the wider the speed range that the vehicle can move along the path without sliding, tipping over, or flying off the ground.

3.4 Global Search and Local Optimization

The search for the optimal path follows the method presented in [4]. It combines a grid search in the position space with a local optimization to yield the global optimal path for a variety of static and dynamic cost functions, such as distance and motion time. This approach eliminates the search in the $2n$ dimensional state-space without sacrificing global optimality.

The cost function for Mars rover is computed by dividing the path length by the maximum constant speed that does not cross the velocity limits for that path. This cost function is the minimum motion time at the constant speed along the path. It quantifies the cumulative effects of path distance, terrain topography, and vehicle dynamics. It also favors regions with high velocity limits, which are traversable at the widest speed range.

The optimization starts by searching for a set of best paths along a uniform grid over the terrain, using the Dryfus algorithm. These paths are pruned by retaining the best path in each neighborhood, each representing the neighborhood of a potential local minimum. Submitting these paths to a local optimization that further minimizes the cost function yields the global optimal path in addition to a set of good alternatives. This optimization, admits paths that might go over obstacles if such a path is dynamically feasible and it is less costly than going around.

4 Comparison of Results

The two planners were tested on images obtained from the JPL Mars Yard. The images were electronically manipulated to make the terrain more challenging by adding large rocks in the central region. A monochrome version of the color image used for path planning is shown in Fig. 2.

In the absence of stereo images, the apparent rock height and size were determined from a single image based on several assumptions on camera location and geometry. The height is estimated by multiplying the apparent height by a correction factor derived from perspective transformation. Similarly the size of a rock is estimated from its apparent boundary by subjecting it to perspective transformation. The the number of pixels within the perspectively corrected boundary is then found, giving the size (area) of the rock. A contour map is then constructed on the basis of location, height and size of each obstacle. The contour map of the Mars terrain (Fig. 2) is shown in Fig. 3, where darker areas correspond to higher elevations. This contour map was used by both path planner.

For the genetic planner, the 512 × 512 pixel image representing a 10 square meter region was divided into 32 × 32 cells. The number of cells can be increased for higher resolution, if required. The impedance of each cell was determined using the method described in Section 2.1. A population size of five paths was chosen, and these paths went through the genetic evolution described in Section 2. The initial intraversable paths were quickly evolved into traversable paths, and as the evolution continued these paths in turn changed into shorter ones passing through less rock concentrated

		Height			
Size	VL	LO	ME	HI	VH
TI	VL	VL	LO	ME	VH
SM	VL	VL	LO	ME	VH
ME	VL	LO	LO	LO	VH
LG	VL	LO	ME	LO	VH
XL	VL	LO	ME	LO	VH

Fig. 1. The fuzzy rule matrix. The entries are terrain roughness

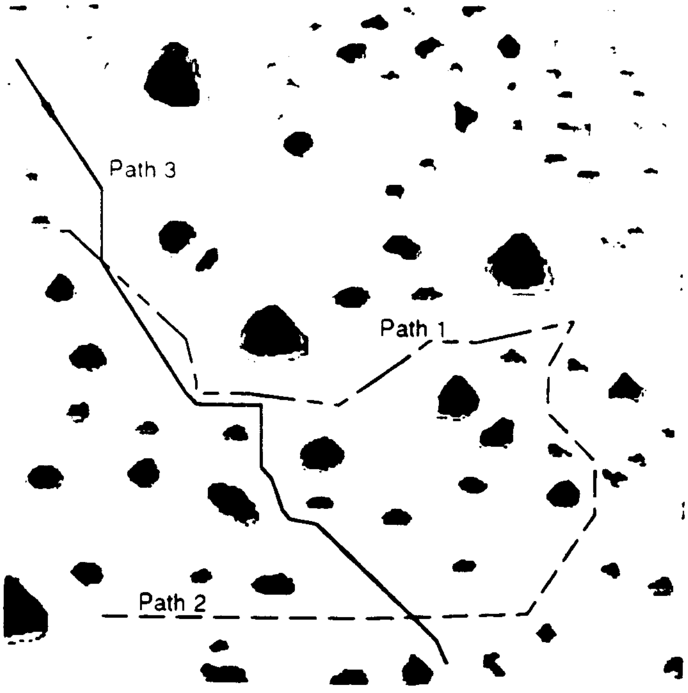

Fig. 3. Three paths found by the genetic path planner shown on the contour map (Smoothing not applied to the paths)

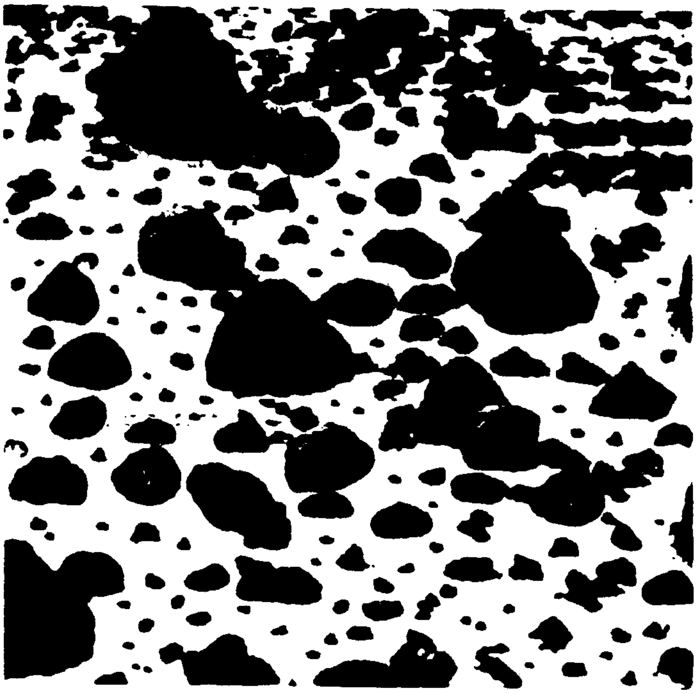

Fig. 2. A reconstructed Mars image

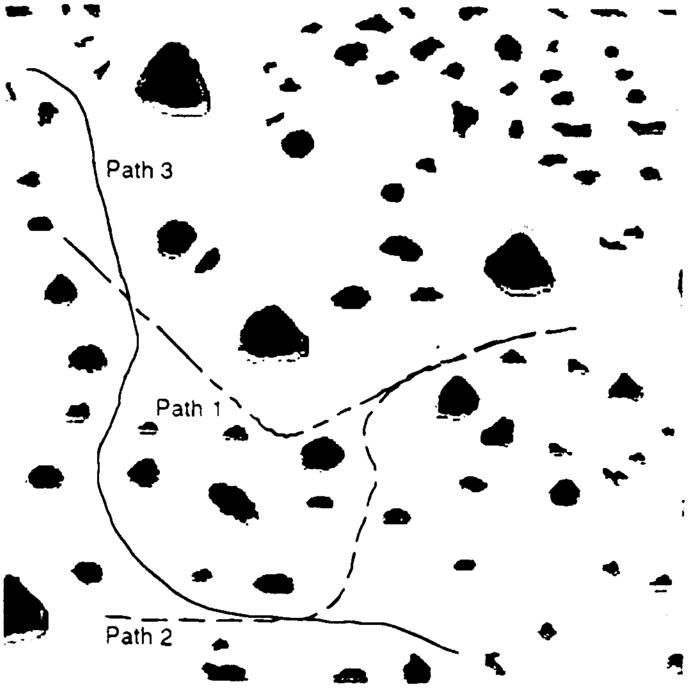

Fig. 4. Three paths found by the global optimization planner shown on contour map

areas and avoiding larger rocks. Near optimal paths were usually found after 200 to 400 iterations (generations), thus good paths were found very quickly. Figure 3 shows three path generated by the genetic algorithm. Path 1 starts at the left part of the region near a rock and the goal position is located to the right of the region at the base of a large rock. Path 2 starts at the lower left corner and has the same goal location as Path 1. Path 3 starts at the upper left corner and has its goal location in the lower center of the region.

The global planner uses the contour map directly, and performs the optimization method described in Section 2. Figure 4 shows the paths found by the global planner for the same start and goal locations as those used for the genetic planner.

Several observations are now made regarding the generated paths. First, the genetic planner produces the waypoints, and in Figure 3 these waypoints are connected by straight line segments. To obtain smoother paths, these waypoints can be connected by cubic polynomials or any other suitable interpolations. It is also noted that in these runs a low weighting (a in (3)) was assigned to curvature relative to the cell impedance to obtain shorter paths. As a result a path sometimes traverses over small rocks to achieve shorter path lengths (and path impedance). However, a closer examination shows that all paths are in fact traversable by the rover (in this case NASA's Rocky 7 rover [2]). The global optimization planner produces smoother path due to using a finer grid resolution.

Even though both planners attempt to optimize their respective performance indices, they have different conceptual basis. The genetic planner employs a fuzzy description of the terrain, and attempts to come up with a path that is short and passes over reasonably smooth parts of the terrain. It delegates the local maneuvering of the rover along the planned path to the rover navigation system. Thus the rover kinematics and dynamics are only considered indirectly through terrain topology during the path planning phase. The global planner uses both terrain topology information and a simplified kinematic/dynamic rover model to achieve both path planning and navigation. As a result of the added task of taking kinematic/dynamic constraints into considerations, it is generally more complex and requires more computation compared to the genetic planner. This added complexity is justified provided that a reasonably accurate terrain topology can be constructed from the images of the terrain, and that the simplified kinematic/dynamic model can adequately represent the actual rover behavior. On the other hand, the genetic planner requires only imprecise information about the terrain but relies upon on-line hazard detection for possible local adjustments to the path. The paths produced by both planners are generally longer than the shortest paths between respective end points (Fig. 3 and 4) but they seem to pass mostly through wider corridors and hence are safer.

5 Conclusions

The path planners described in this paper share the common attribute of attempting to optimize certain performance indices. It has been shown through planning of paths for a simulated Mars terrain that both are capable of producing short paths that traverse over smooth parts of the terrain and avoid areas with large rocks. While both planners perform some form of optimization, they are conceptually different. The genetic planner requires only an approximate description of the terrain and operates on the basis of evolutionary process and stochastic search to generate a near optimal path. The global planner incorporates certain kinematics and dynamics into the planning phase, and require more knowledge about the environment and the rover. The relative simplicity of the genetic planner and the benefit of incorporating kinematic/dynamic constraints of the global planner can be combined to achieve better results. For example, the genetic planner can quickly produce a number of paths based on imprecise terrain description and the global planner can then evaluate or modify these paths to take into consideration the rover kinematic/dynamic constraints.

6 References

[1] J.-C Latomb, Robot Motion Planning, Kluwer Academic Publishers, 1991.

[2] S. Hayati et al, "The Rocky 7 rover: A Mars sciencecraft prototype," Proc. IEEE Int. Conf. Robotics and Automation, pp. 2458-2464, Albuquerque, NM, April 1997.

[3] S. Laubach, J. Burdick and L. Matthies, "A practical autonomous path planner implemented on the Rocky 7 prototype microrover," IEEE Int. Conf. Robotics and Automation, 1998.

[4] Z. Shiller and Y-R. Gwo, "Dynamic motion planning of autonomous vehicles," IEEE Trans. Automation and Robotics, pp. 241-249, vol. 2, 1991.

[5] T. Simeon and B. Darce-Wright, "A practical motion planner for all-terrain mobile robots," Proc. Int. Conf. Intelligent Robots and Systems, 1993.

[6] S. Farritor, H. Hacot, and S. Dubowsky, "Physics-based planning for planetary exploration," IEEE Int. Conf. on Robotics and Automation, 1998.

[7] M. Tarokh, R. Chan and C. Song, " Path planning of rovers in rugged terrain using fuzzy logic and genetic algorithm," Technical Report, Robotics and Intelligent Systems Laboratory, San Diego State University, 1999.